

REMARKS

Claims 1-83 are pending in this application. Claims 10, 12, 55, and 82 have been canceled. Applicant has amended claims 1-3, 9, 11, 15, 16-17, 24-26, 31-32, 36-37, 40-42, 47-50, 54, 62-63, 66, 72, 76, 78-81, and 83 to more particularly point out and distinctly claim applicant's invention and to address typographical and stylistic errors. No new matter has been introduced by way of these amendments.

Please note that several of applicant's claims, including claims 15-17, 26, 31, 40-42, 49, 62, 63, and 76, have been amended to recite language that under recent case law is intended to state two or more choices in the alternative and to clarify that the previously recited language is not to be interpreted solely in the conjunctive. Thus applicants have amended claims 15-17, 26, 31, 40-42, 49, 62, 63, and 76 to substitute the disjunctive "or" for "and" before the last element in each alternative list so that the respective claim covers any one or more of the listed elements in the alternative. Namely, the phrase "at least one of A or B" is to be interpreted as including: "A," or "B," or both "A and B." Also, the phrases "at least one of A, B, or C," "A, B, or C," and "A or B" are to be interpreted in the same manner. These amendments conform the claims with current case law on this subject.

Rejections Under 35 U.S.C. § 102 and § 103

The Examiner has rejected claims 1-24, 32-47, 54-55, 58-67, 69-70, 72, 74-79 and 81-83 under 35 U.S.C. § 102(e) as anticipated by U.S. Patent No. 6,016,553 issued to Schneider, et al. ("Schneider"). In addition, the Examiner has rejected claims 73 under 35 U.S.C § 103(a) as obvious over Schneider in view of the Examiner's assertions of what is known. The Examiner has also rejected claims 25-31, 48-53, 56-57, 68, 71 and 80 under 35 U.S.C § 103(a) as being obvious over Schneider in view of White et al. ("White"), U.S. Patent No. 6,092,161 (of record). Applicant respectfully traverses the Examiner's rejections.

In essence, the Examiner appears to be asserting that Schneider's described backup (data restoration) techniques, which provide a user with a means for (explicitly) "rolling back the state of a disk to a previous time ... by maintaining original and current states and a mapping system to guide how these should be combined to create a given state corresponding to

some specific time in the past”, somehow teach, suggest, or motivate applicant’s claimed redirecting techniques that preserve data so that, for example, as recited in claim 1 before amendment, “the data stored in the original location ... remains unaltered, without any restorative copying of the data.” (See Schneider, column 12, lines 45-50, hereinafter indicated in col#:line# format.) More specifically, on page 4 of the Office Action, the Examiner appears to indicate that, because Figure 13A shows an initial (empty) state of the extra pages of the Schneider engine in which backup history is not yet available (since the engine hasn’t yet run, no changes to the disk have been made), then, if the computer system is restarted (or rebooted) at that point, no restorative copying will be done. Therefore, presumably the Examiner is asserting a conclusion that Schneider somehow teaches, motivates, or suggests specifically using the engine to achieve such a state or that it can even be used to do so.

Applicant respectfully disagrees for several reasons. Applicants claimed methods, systems, and computer readable memory mediums provide techniques for “automatically preserving an original state of a computer system upon rebooting” by providing a redirection driver *that does not alter protected data*. (See Applicant’s Specification, hereinafter “Spec”, p. 2, line 15.) Applicant’s techniques allow an entity such as a library to provide a computer system for potentially uncontrolled use, such as might occur in a public or multi-user setting, yet be reassured that the computer system cannot be corrupted because it can simply be rebooted to return the system to an initial state. (See Spec, p.1, lines 14-23.)

Applicant’s claims recite several aspects that are nowhere taught, suggested, or motivated by Schneider. Specifically, each of independent claims 1, 2, 3, 32, 72, and 79 before amendment, and by virtue of incorporation all of the pending dependent claims, recite in some manner that, after redirecting access requests, “the data in the [original] location *remains unaltered*” when the computer system is rebooted/restarted (emphasis added). Or, as recited in claim 54 before amendment, and, virtue of incorporation dependent claims 56-71, “when the computer system is rebooted..., locations in the protected space *remain unaltered*” (emphasis added). Although the claims have been amended for clarity to emphasize that protected data remains unaltered “*automatically*,” after a (write) access request has been intercepted and a modification made to a redirected area, applicant believes that these claims prior to amendment

also included acts or elements not taught, suggested or motivated by Schneider. However, in furtherance of prosecution, (and not to be interpreted as agreement with the Examiner's position) applicant will focus hereafter on the claim language as amended.

As amended, claim 1 recites,
under control of the redirection driver,
...
redirecting the write access request ... such that the request transparently writes to the current redirected location instead of the original location; and
restarting the computer system from a second powered-down state, wherein the data stored in the plurality of original locations on the storage device ***automatically remains unaltered from the original state***, without restorative copying of data to the plurality of original locations.

(emphasis added).

Similarly, independent claim 2, as amended, recites,
... redirects the access request ... such that the requested modification at the original location is not performed and is instead performed to the redirected location, and such that, when the computer system is restored ... the data in the original location on the storage device ***automatically remains unaltered from the original state*** without any restorative copying of data to the plurality of original locations.

(emphasis added).

Independent claims 3 and 32, as amended, and by virtue of incorporation, claims 4-9, 11, 13-31 and 33-53 recite,

... redirecting the intercepted request to modify the determine location ... so that the data stored in the location in the protected space ***automatically remains unaltered*** when the computer system is restarted from a powered-down state.

(emphasis added).

Independent claim 54, as amended, and by virtue of incorporation, claims 56-71 recite,

... redirects intercepted requests so that the requests result in modifying locations in the redirected storage space instead... thereby leaving the protected space unaltered so that, when the computer system is rebooted... the locations in the protected space ***automatically remain unaltered*** without restorative copying of data to the protected space.

(emphasis added).

Independent claim 72, as amended, and by virtue of incorporation, claims 73-78 recite,

... redirecting the request to modify a location ... such that the data in the location... remains unaltered;
restarting the computer system from a powered-down state, wherein the data stored in the location on the storage device *automatically remains unaltered*, without requiring any restorative copying of data to the location on the storage device.

(emphasis added).

Independent claim 79, as amended, and by virtue of incorporation, claims 80, 81, and 83 recite,

... redirects the write request to the associated redirected location so that data in the designated location *automatically remains unaltered* even when the computer system is rebooted.

(emphasis added).

There is no teaching in Schneider of redirecting write access to a disk in a manner that preserves the original content of the protected data such that when restarting or rebooting the computer system, “the [original] data [or locations]... *automatically remains unaltered*” (emphasis added). In contrast, Schneider attempts to address a completely different and opposite problem – that of *backing up* a disk whose data is being altered on a continuous basis by historically tracking changes to the disk so that a user can (explicitly) revert the disk to an earlier point in time as chosen. (See Schneider, 2:32-35.) This is not the same as automatically keeping data unaltered as claimed. Thus, the Examiner’s conclusion that Fig. 13A teaches applicant’s recited claim language, doesn’t logically follow from Fig. 13A, because once the Schneider engine is started (which is necessary to accomplish the recited aspect of “intercepting a data ...request” to modify some location on a disk), then the history buffer (the “extra pages”) will not be empty and some data will require restorative copying prior to the reversion to an initial state without modifications.

Moreover, there does not appear to be any teaching, motivation, or suggestion in Schneider to use the backup system as the Examiner suggests to *avoid modifying* original data in order to provide a reboot that keeps protected data unchanged, yet allows writing to the disk as if the original data were being modified. Rather, Schneider suggests the opposite, when it explains at great length how all of the incremental modifications to the disk, while initially perhaps diverted to other locations, are eventually updated (swapped) to the target locations in the background, while a history of changes is being kept in the history buffer. (See Schneider, for

example, 17:32-44, “The approach is to generally walk through the map, swapping pages back where they really belong.”)

In contrast to preventing disk modifications so that the original data “automatically remains unaltered,” as recited by applicant’s claims, the Schneider system describes a sort of progressive “undo” capability for a disk because it allows modifications. As stated in Schneider, “the present invention provides a method and apparatus for *information recovery* focusing, in one example embodiment, on the second situation... where information is *altered* and access to its original state may be desired.” (Schneider, 1:49-50 (emphasis added).) Schneider describes techniques “for reverting a disk drive to an earlier point in time” using a circular history buffer to provide a “continuously running disk backup system.” (Schneider, Abstract; 2:32-35.) In particular, the history buffer can provide a history of “old data, the time it was replaced by new data, and the original location of the data....” or it can be used to store “new data elements ... leaving the old data elements in their original locations.” (Schneider, Abstract.) In either case, an extensive mapping and swapping system is employed by the engine to eventually (typically during background processing) swap the newest data into its actual disk locations and put older historic data into a history buffer controlled by a map. Depending upon the particular Schneider method used, the history buffer contents may reside as a combination of data in the main area and data in the extra area at some particular point in time. (See, *e.g.*, Figs. 2, 6, 13D.)

In Schneider, to restore the disk to an earlier state, even after a system crash or reboot, a user explicitly chooses a safe point to which to restore the data, and potentially a list of files to restore. (See Schneider, for example, 12:62-63. “The user is allowed to select only a safe point in time to which to revert.”; *see also*, 61:42-62:10.) Further, “[i]n the event of a crash, it is assumed the disk would come back up in its state *as of the last safe point*.” (Schneider, 14: 2-4 (emphasis added).) Thus, the state the system boots into after a system crash (before any user reversion) is its last (safe) state – which is suggested to be a point during which the engine is already running since “safe” is a concept defined and implemented by the engine– not an initial state of the disk before any modifications took place (and before the engine is run, as in Fig. 13A). The concept of a safe point refers to a point at which the disk writes are not in transition,

for example, before or after a file is written out to disk, but not somewhere in the middle. (See, e.g., 12:50-42.) There appears to be nothing in Schneider that teaches a reboot to explicitly keep the data unaltered.

Thus, at least one aspect of claims 1-9, 11, 13-53, 56--81, and 83 is not taught, suggested or motivated by Schneider, therefore, for these reasons and others, claims 1-9, 11, 13-53, 56-81, and 83 are not anticipated by or obvious in view of Schneider.

The Examiner has also rejected several of the dependent claims as obvious over Schneider in view of White. Specifically, the Examiner has rejected claims 25-31, 48-53, 56-57, 68, 71, and 80, and has admitted that Schneider does not teach "receiving a request to shutdown ... and the subsequent action of disregarding the data in the redirected area;" "receiving a request to shutdown ... and the subsequent action of saving the data in the redirected area;" "using redirection tables to associate locations in the protected space to locations in the redirected space;" or "the redirection driver disregarding request to access locations referred to by the unprotected space." The Examiner then asserts that White provides these missing aspects and states that the motivation to combine each such aspect of White with Schneider is that "it would have been obvious to one of ordinary skill[sic] in the art ... to *realize the benefit of* <whatever the aspect recites> ... and to incorporate it into the existing scheme disclosed by Schneider et al. to *further enhance the flexibility of the system.*" (Page 12, Office Action.)

As a preliminary matter, the Examiner hasn't met his *prima facie* burden of establishing that Schneider and White can be properly combined in an obviousness rejection. Merely stating that it would be obvious to have some feature of one reference in the other reference to "allow further enhance the flexibility..." does not meet this burden -- there must be some teaching or suggestion *in either or both references or in the art* to legitimately combine them. The Examiner has pointed to no such suggestion. Further, hindsight reconstruction, using applicants' claims as a template for combining the references is also not permitted.

First, White and Schneider are directed to solving different problems, and thus there is no apparent reason on the surface to combine them. Schneider is directed, on the one hand, to software for making and tracking modifications for backup purposes (see Schneider, 2:32-35); while White is directed to hardware/firmware that resides between a PC and a disk, for

restricting modifications (by allowing temporary modifications that are discarded) to pre-designated areas on a disk, such as boot partitions (see White, Figs. 6-8, 1:65-2:32).

Second, several of such combinations would render Schneider inoperable. For example, as per claims 25-26 and 48-49, the Examiner asserts that White teaches that using a system reset for “disregarding of the updated data in the redirected area saves disk space and is easy to implement.” However, Schneider teaches away from preserving the overwritten (*i.e.*, original) data in one area. Schneider describes instead using a combination of main pages and extra pages to store original and new data and swapping data/map pointers in the background on a continuous basis. (See *e.g.*, Schneider, 17:32-44.) At any one point in time the new data is not likely to be found in one area. If one were to modify Schneider to disregard all of the data in a “redirected” area upon reset (for example, any area in the engine to which “writes” are diverted), the engine is likely to be in some intermediate state – where some amount of original data has been disregarded as well as some new data, and where some amount of new data is kept as well as some original data. Thus, such a combination would not enhance the backup capabilities of Schneider, but would rather confuse them. Similar inoperability arguments can be applied to the other aspects of White in combination with Schneider, and are omitted for brevity.

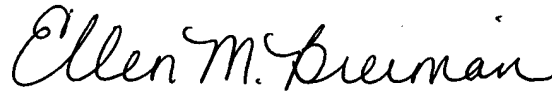
Therefore, for these reasons and others, neither Schneider nor White, alone or in any motivated combination teach, motivate or suggest one or more aspects of the independent claims and thus claims 1-3, 32, 54, 72, and 79 are not anticipated by or rendered obvious in view of Schneider and/or White. Similarly, because dependent claims 4-9, 11, 13-31, 33-53, 56-71, 73-78, 80, 81, and 83 incorporate these aspects by virtue of their dependencies, claims 4-9, 11, 13-31, 33-53, 56-71, 73-78, 80, 81, and 83 also are not anticipated by or rendered obvious in view of Schneider or White, alone or in any motivated combination, for at least the reasons set forth above.

In closing, applicant respectfully submits that all of the pending claims are thus allowable and respectfully requests the Examiner to enter these amendments and to reconsider this application and its timely allowance. In the event the Examiner disagrees with applicant or finds minor informalities, applicant invites the Examiner to contact applicant’s representative to

discuss the Examiner's issues and to expeditiously resolve prosecution of this application. Applicant's representative can be contacted at (206) 622-4900. The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090. Again, applicant's representative thanks the Examiner for his prompt and courteous attention.

Respectfully submitted,

SEED Intellectual Property Law Group PLLC

A handwritten signature in cursive script that reads "Ellen M. Bierman". The signature is written in dark ink and is positioned above a horizontal line.

Ellen M. Bierman
Registration No. 38,079

EMB:asl

Enclosure:
Postcard

701 Fifth Avenue, Suite 6300
Seattle, Washington 98104-7092
Phone: (206) 622-4900
Fax: (206) 682-6031

585994_1.DOC